Coursera - Machine Learning - Peer-graded Prediction Assignment

Elaparthi

Priyanka

05 May 2024

Overview

The goal of this project is to predict the manner in which 6 participants did various exercises. There is a “classe” variable available in the training set. I have created a report describing how I built my model and used cross validation, what I think the expected out of sample error is, and why I made the choices I had to. I will also use my prediction model to predict 20 different test cases.

Summary

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. More information is available from the website here: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

Data Source

The training data for this project is available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv)

The test data is available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har). I would like to specially thank following persons for their permission to allow me to use their data for my assignment. Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human ’13) . Stuttgart, Germany: ACM SIGCHI, 2013.

Download, Read and Clean Data

Training Data

```
#setwd("/Users/yogeshdhar/Coursera/Machine Learning Project") # Set working directory

trainURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
# Download training data and save it

train <- read.csv(trainURL, na.strings=c('','NA'), header = TRUE) # We can see that our dataset has 19622 observations of 160 variables

train_clean_NA <- train[,!apply(train,2,function(x) any(is.na(x)) )] # We now have 19622 observations of 60 variables after removing NA values

train_final=train_clean_NA[,-c(1:7)]
# We have removed comumns 1:7, as they are non numeric in nature abd won't be significant for our analysis. This leaves us with 19622 observations of 53 variables
```

Testing Data

```
testURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
# Download test data and save it

test <- read.csv(testURL, na.strings=c('','NA'), header = TRUE) # We can see that our dataset has 20 observations of 160 variables

test_clean_NA <- test[,!apply(test,2,function(x) any(is.na(x)) )] # We now have 20 observations of 60 variables after removing NA values

test_final=test_clean_NA[,-c(1:7)]
# We have removed comumns 1:7, as they are non numeric in nature and won't be significant for our analysis. This leaves us with 20 observations of 53 variables
```

List of field names being used for analysis

```
colnames(test_final)
```

```
##  [1] "roll_belt"             "pitch_belt"            "yaw_belt"
##  [4] "total_accel_belt"      "gyros_belt_x"          "gyros_belt_y"
##  [7] "gyros_belt_z"          "accel_belt_x"          "accel_belt_y"
## [10] "accel_belt_z"          "magnet_belt_x"         "magnet_belt_y"
## [13] "magnet_belt_z"         "roll_arm"              "pitch_arm"
## [16] "yaw_arm"               "total_accel_arm"       "gyros_arm_x"
## [19] "gyros_arm_y"           "gyros_arm_z"           "accel_arm_x"
## [22] "accel_arm_y"           "accel_arm_z"           "magnet_arm_x"
## [25] "magnet_arm_y"          "magnet_arm_z"          "roll_dumbbell"
## [28] "pitch_dumbbell"        "yaw_dumbbell"          "total_accel_dumbbell"
## [31] "gyros_dumbbell_x"      "gyros_dumbbell_y"      "gyros_dumbbell_z"
## [34] "accel_dumbbell_x"      "accel_dumbbell_y"      "accel_dumbbell_z"
## [37] "magnet_dumbbell_x"     "magnet_dumbbell_y"     "magnet_dumbbell_z"
## [40] "roll_forearm"          "pitch_forearm"         "yaw_forearm"
## [43] "total_accel_forearm"   "gyros_forearm_x"       "gyros_forearm_y"
## [46] "gyros_forearm_z"       "accel_forearm_x"       "accel_forearm_y"
## [49] "accel_forearm_z"       "magnet_forearm_x"      "magnet_forearm_y"
## [52] "magnet_forearm_z"      "problem_id"
```

Model Building and Assessment

Load Required Packages

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.4.4
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.4
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##     margin
```

```
library(rpart)
```

Data Partition (Training and Testing dataset)

```
inTrain <- createDataPartition(y=train_final$classe, p=0.7, list=FALSE)
Training <- train_final[inTrain,]
Testing <- train_final[-inTrain, ]
dim(Training);dim(Testing)
```

```
## [1] 13737    53
```

```
## [1] 5885   53
```

```
# We have 13737 observations in the Training dataset and 5885 observations in Testing datase
t.
```

Approach to Prediction models

For this project I have used FOUR different models to figure out the one with maximum accuracy (prediction rate) out of sample dataset. Following are the models I am going to use:

Recursive Partitioning and Regression Trees (rpart) Generalized Boosted Models (gbm) Random Forest Decision Trees (rf) Boosted Logistic Regression (LogitBoost)

Cross validation

I have applied Cross validation for each model with K = 3 by using the trainControl object as defined below. This method has been used to control the computational nuances

```
train_control <- trainControl(method="repeatedcv", number=3)
```

The codes required to build our predictive models are given below:

Method = Recursive Partitioning and Regression Trees (rpart)

```
set.seed(1000)

model_Fit_rpart <- train(classe ~ ., data = Training, trControl = train_control, method = "rp
art")

Predict_model_rpart <- predict(model_Fit_rpart,Testing, type="raw")

Confusion_Matrix_rpart <- confusionMatrix(Predict_model_rpart,Testing$classe)

save(Confusion_Matrix_rpart,file='testrpart.RData')
load('testrpart.RData')

Confusion_Matrix_rpart$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1529  500  477  443  154
##          B   30  374   38  150  156
##          C  110  265  511  371  309
##          D    0    0    0    0    0
##          E    5    0    0    0  463
```

```
Confusion_Matrix_rpart$overall[1]
```

```
## Accuracy
##  0.48887
```

Method = Generalized Boosted Models (gbm)

```
set.seed(1000)

model_Fit_gbm <- train(classe ~ ., data = Training, trControl = train_control, method = "gbm"
, verbose = FALSE)

Predict_model_gbm <- predict(model_Fit_gbm,Testing, type="raw")

Confusion_Matrix_gbm <- confusionMatrix(Predict_model_gbm,Testing$classe)

save(Confusion_Matrix_gbm,file='testgbm.RData')
load('testgbm.RData')

Confusion_Matrix_gbm$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1649   53    0    3    2
##          B   17 1037   31    5   12
##          C    6   38  977   37   13
##          D    1    7   17  916   14
##          E    1    4    1    3 1041
```

```
Confusion_Matrix_gbm$overall[1]
```

```
##  Accuracy 
## 0.9549703
```

Method = Random Forest Decision Trees (rf)

```
set.seed(1000)

model_Fit_RF <- train(classe ~ ., data = Training, trControl = train_control, method = "rf",
 verbose = FALSE)

Predict_model_RF <- predict(model_Fit_RF,Testing, type = "raw")

Confusion_Matrix_RF <- confusionMatrix(Predict_model_RF,Testing$classe)

save(Confusion_Matrix_RF,file='testrf.RData')
load('testrf.RData')

Confusion_Matrix_RF$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1669    8    0    0    0
##          B    3 1127    4    1    0
##          C    1    4 1019   16    2
##          D    0    0    3  947    0
##          E    1    0    0    0 1080
```

```
Confusion_Matrix_RF$overall[1]
```

```
## Accuracy 
## 0.9926933
```

Method = Boosted Logistic Regression (LogitBoost)

```
set.seed(1000)

model_Fit_LogitBoost <- train(classe ~ ., data = Training, trControl = train_control, method = "LogitBoost", verbose = FALSE)

Predict_model_LogitBoost <- predict(model_Fit_LogitBoost,Testing, type = "raw")

Confusion_Matrix_LogitBoost <- confusionMatrix(Predict_model_LogitBoost,Testing$classe)

save(Confusion_Matrix_LogitBoost,file='testLogitBoost.RData')
load('testLogitBoost.RData')

Confusion_Matrix_LogitBoost$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1473   82   11   19    8
##          B   36  796   47    9   46
##          C    6   45  636   24    5
##          D   14   18   64  737   32
##          E   11   17    7   16  873
```

```
Confusion_Matrix_LogitBoost$overall[1]
```

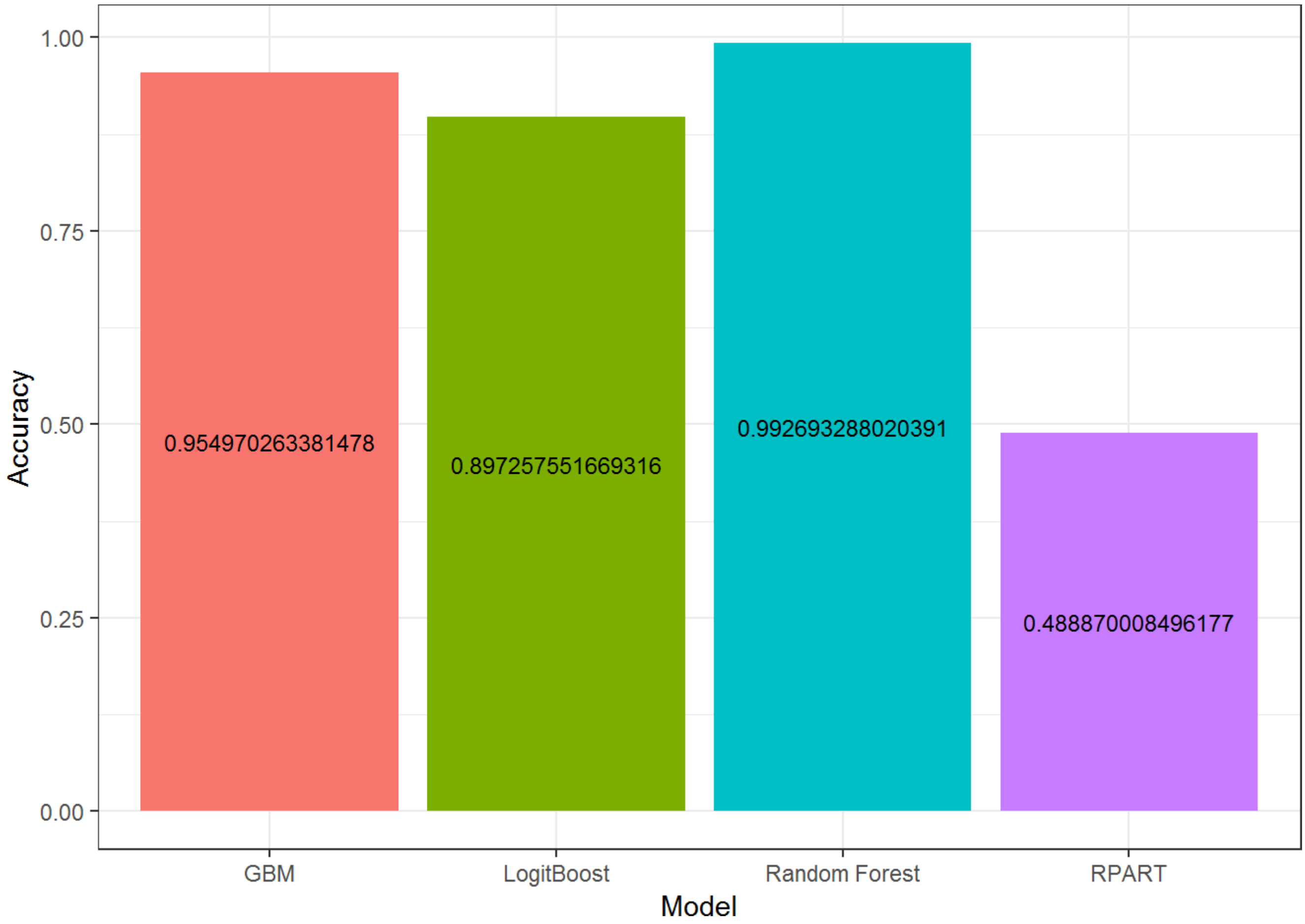
```
##  Accuracy 
## 0.8972576
```

Visualising Model performances and Concluding

```
Predicted_values <- c(Confusion_Matrix_rpart$overall[1], Confusion_Matrix_gbm$overall[1], Confusion_Matrix_RF$overall[1],Confusion_Matrix_LogitBoost$overall[1])

Models_list <- c("RPART", "GBM", "Random Forest", "LogitBoost")

data_frame <- data.frame(Model = Models_list,
                 Accuracy = Predicted_values)
ggplot(data_frame, aes(x = Model, y = Accuracy)) + geom_bar(stat = "identity", aes(fill = Model)) + theme_bw() + theme(legend.position = "none") + geom_text(aes(label = Predicted_values), size = 3, position = position_stack(vjust = 0.5))
```



Post reviewing various model performances, we can conclude that Random forest is the best performing model, followed by Generalized Boosted Model. Hence, we Will use the Random Forest model for this assignment.